

本周周报（12.1-12.7）

刘昊南

本周工作

1. 开放了 CS 版本中地形高度的 `scale`，现在可以在程序的设置界面中调整 `dem` 的 `scale` 值，相应的场景中的地形高度能够实时的改变
2. 按照气象局的需求，修改了气压层高度，不在按实际值绘制。各气压层的高度差相等，能够调整，且最低层的高度也可以调整，这样就可以供用户调整出一个比较好的可视化效果。目前在调整为新的参数后添加的可视化对象中气压层高度是按新的参数绘制的，尚不清楚是否需要之前添加的可视化对象也做高度上的调整，要做这样的效果需要调整所有的 `vertex shader`，要麻烦一些。
3. 考虑到需要做实时拾取的功能的话，就需要改变使用 `CGI` 请求在服务器端拾取的做法，改为在客户端下载数据块进行拾取，并实现数据在内存中的 `LRU` 管理功能。
4. 仔细思考了目前 CS 端程序的数据获取方式，现在要获取一个数据块会先检查数据是否在硬盘中，否则下载该数据。现在需要改为，先检查数据是否缓存在内存中，然后检查是否在硬盘中，最后下载数据。因此，程序的架构有必要作出改变，实现一个类似 Web 版中的 `ResourceManager` 负责数据块的缓存、读取以及下载，以 `LRU` 的方式进行统一管理。
5. CS 端相对 web 版十分流畅的原因在于使用了多线程，开辟了一个单独的线程用户数据的读取，一个单独的线程用于地球 `mesh` 的计算，因此需要将这两个线程与 `ResourceManager` 整合到一起，`ResourceManager` 位于主线程方便各个可视化模块的调用。
6. `ResourceManager` 与两个 `worker thread` 之间的通信的方式有两种，一种是 `ResourceManager` 之间调用 `worker thread` 中的 `method` 来发出请求，`worker thread` 先返回一个 `state` 为 `false` 数据块，`ResourceManager` 轮询该数据块知道该数据块的 `state` 为 `true`；另一种方式是采用 `signal-slot` 的方式，`ResourceManager` 通过 `signal` 发出请求，`worker thread` 在完成数据的获取后也通过 `signal` 的方式将数据块返回给 `ResourceManager`，这得益于 `worker thread` 和主线程由单独的 `event loop`。目前我们的实现是类似第一种的方式，不过是各个单独的下载类进行调用的，在调用 `worker thread` 的 `method` 时，需要引入互斥锁，以保护 `worker thread` 中请求队列的完整性，效率较差。因此，准备改为第二种方式，这样不会有损害 `worker thead` 数据完整性的风险，同时有较高的效率，也利于回调机制的设计。

下周计划

1. 先完成截屏以及录视频的功能
2. 完成 ResourceManager